

# Accelerating linear model predictive control by constraint removal

Michael Jost<sup>a</sup>, Gabriele Pannocchia<sup>b</sup>, Martin Mönnigmann<sup>a,\*</sup>

<sup>a</sup>*Automatic Control and Systems Theory, Ruhr-Universität Bochum, Universitätsstraße 150, 44801 Bochum, Germany*

<sup>b</sup>*Department of Civil and Industrial Engineering, University of Pisa, Largo Lucio Lazzarino, 2, 56126 Pisa, Italy*

---

## Abstract

Model predictive control (MPC) is computationally expensive, because it is based on solving an optimal control problem in every time step. We show how to reduce the computational cost of linear discrete-time MPC by detecting and removing inactive constraints from the optimal control problem. State of the art MPC implementations detect constraints that are inactive for all times and all initial conditions and remove these from the underlying optimization problem. Our approach, in contrast, detects constraints that become inactive as a function of time. More specifically, we show how to find a bound  $\sigma_i^*$  for each constraint  $i$ , such that a Lyapunov function value below  $\sigma_i^*$  implies constraint  $i$  is inactive. Since the bounds  $\sigma_i^*$  are independent of states and inputs, they can be determined offline. The proposed approach is easy to implement, requires simple and affordable preparatory calculations, and it does not depend on the details of the underlying optimization algorithm. We apply it to two sample MPC problems of different size. The computational cost can be reduced considerably in both cases.

*Keywords:* model predictive control, linear systems, constrained control, quadratic programming

---

## 1. Introduction

Model predictive control (MPC) is a powerful method for the control of constrained, multivariable systems. Because MPC requires to solve optimal control problems online, it is computationally expensive, however. For the common case of linear systems, linear constraints and quadratic objective functions, the problem to solve is a Quadratic Program (QP), and the control law implicitly

---

\*Corresponding author. Tel.: +49 234 32 24060

Email addresses: [michael.s.jost@rub.de](mailto:michael.s.jost@rub.de) (Michael Jost), [gabriele.pannocchia@unipi.it](mailto:gabriele.pannocchia@unipi.it) (Gabriele Pannocchia), [martin.moennigmann@rub.de](mailto:martin.moennigmann@rub.de) (Martin Mönnigmann)

defined by the MPC problem is known to be a continuous piecewise affine function of the state [1]. Unfortunately, an explicit formula for the control law can only be calculated for very simple problems. Therefore, the online solution of the QP is usually the *only* available option for many problems. Tailored algorithms to solve online this specific QP have been developed over the last decades. For instance, Rao et al. [2] proposed a Riccati recursion based strategy to efficiently solve the KKT system of a primal-dual interior-point method. Wang and Boyd showed how early termination of interior-point iterations and suitable warm-start techniques can be combined to achieve fast MPC computation [3]. Recently several authors investigated the use of Nesterov’s fast gradient method (see e.g. [4]) and the Alternating Direction Method of Multipliers (see e.g. [5]).

It is the purpose of the paper to present a method that uses, very generally speaking, the insights into the structure of the state feedback law without ever determining it. We argue this idea is appealing, because it applies to a much larger class of problems than those for which the explicit law can actually be calculated. We briefly summarize some related publications that are also based on this general idea but use techniques different from the one proposed here. Ferreau et al. [6] show that the piecewise structure of the explicit control law can be exploited to accelerate the online computations of MPC.<sup>1</sup> Zeilinger et al. [7] demonstrate how to use a suboptimal approximation of the piecewise-affine control law to warm-start the online optimization problem and thus speed up its solution. Pannocchia et al. [8, 9] store the active sets which occur most frequently and use this information to predict the subsequent active sets.

The approach proposed here is based on removing inactive constraints from the online optimization problem before solving it, thus reducing the online computational effort. We stress we do not merely remove constraints that are redundant because they are inactive for all times and all initial conditions. While redundant constraints of this type are captured by the proposed method, a more considerable reduction of the online computation time results from detecting constraints that become inactive along trajectories of the closed-loop system as a function of time.

The idea to detect and remove inactive constraints before solving the online optimization problem has first been proposed by the authors in [10, 11]. These first attempts were based on identifying the *region of activity* of each constraint, or approximations thereof. In the present paper, we show how to detect inactive constraints by exploiting the fact that the objective function is a Lyapunov function for the controlled system under typical assumptions (see also [12]). The method proposed here differs from the ones in [10, 11, 12] in several aspects. In contrast to [10, 11] we do not need to calculate regions of activity or approximations thereof. Instead, we use lower bounds on the objective function to determine if a constraints is inactive or not. Consequently, the preparatory calculations required for the presented approach are much simpler

---

<sup>1</sup>Let MPC refer to the non-explicit case, i.e. MPC based on repeatedly solving optimization problems online.

than those required in [10, 11]. This allows the application of constraint removal to a broader problem class, including control problems that involve large systems and require long horizons.

The method proposed in [12] also uses the optimal cost function to remove inactive constraints. While lower bounds on the optimal cost function are used here, geometric properties are used in [12]<sup>2</sup>. In contrast to [12], we here have to carry out preparatory calculations offline before starting the MPC controller. However, the proposed criterion used to detect inactive constraints online is much simpler than the ones used in [10, 11, 12] from a computational point of view. In fact, only two real numbers have to be compared here, while membership tests and inner products are required in [10, 11] and [12], respectively. In addition, if a constraint is detected to be inactive with the method proposed here, it remains inactive for all times. This is in general not the case for the criteria proposed in [10, 11, 12]. Note that we summarize the method from [10, 11] in Sec. 3.2 in order to be able to compare it to the one proposed here.

The problem class is stated concisely in Sec. 2. Main results are presented in Sec. 3 and applied to two examples in Sec. 4. Conclusions and an outlook are presented in Sec. 5.

## 2. Problem statement

We consider linear discrete-time state-space systems

$$x(t+1) = Ax(t) + Bu(t), \quad (1)$$

with state variables  $x(t) \in \mathbb{R}^n$ , input variables  $u(t) \in \mathbb{R}^m$  and stabilizable pairs  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times m}$ . Assume (1) is subject to the constraints

$$x(t) \in \mathbb{X} \subset \mathbb{R}^n, \quad u(t) \in \mathbb{U} \subset \mathbb{R}^m,$$

for all  $t$ , where  $\mathbb{U}$  and  $\mathbb{X}$  are compact full dimensional polytopes that contain the origin in their interiors. We regulate system (1) to the origin by solving the optimal control problem, on a receding horizon, given the current state  $x$ :

$$\begin{aligned} \min_{X, U} \quad & l_f(x(N)) + \sum_{i=0}^{N-1} l(x(i), u(i)) \\ \text{s. t.} \quad & x(k+1) = Ax(k) + Bu(k), \quad k = 0, \dots, N-1, \\ & x(k) \in \mathbb{X} \subset \mathbb{R}^n, \quad k = 0, \dots, N-1, \\ & u(k) \in \mathbb{U} \subset \mathbb{R}^m, \quad k = 0, \dots, N-1, \\ & x(0) = x, \\ & x(N) \in \mathbb{X}_f, \end{aligned} \quad (2)$$

---

<sup>2</sup>More precisely, we show in [12] that the optimal input sequence for the *successor* state lies in an ellipsoid which is defined by the *current* state alone. This ellipsoid implies a bound on the norm of the optimal input sequence which can be used to detect inactive constraints, however.

where  $X = (x'(1), \dots, x'(N))'$ ,  $U = (u'(0), \dots, u'(N-1))'$ ,  $l(x, u) = \frac{1}{2}x'Qx + \frac{1}{2}u'Ru$  with  $R \in \mathbb{R}^{m \times m}$ ,  $R \succ 0$ ,  $Q \in \mathbb{R}^{n \times n}$ ,  $Q \succ 0$ , and  $l_f(x) = \frac{1}{2}x'Px$  with  $P \in \mathbb{R}^{n \times n}$ ,  $P \succ 0$  solution to the algebraic Riccati equation:

$$P = Q + A'PA - A'PB(B'PB + R)^{-1}B'PA \quad (3)$$

The terminal set  $\mathbb{X}_f \subset \mathbb{X} \subset \mathbb{R}^n$  is assumed to be full dimensional, polyhedral, and to contain the origin in its interior. Furthermore,  $\mathbb{X}_f$  is assumed to be positively invariant for the unconstrained closed-loop system  $x(t+1) = (A + BK)x(t)$  and constraint admissible, i.e.

$$(A + BK)x \in \mathbb{X}_f, \quad Kx \in \mathbb{U}, \quad \text{for all } x \in \mathbb{X}_f, \quad (4)$$

where  $K = -(B'PB + R)^{-1}B'PA$  is the unconstrained LQR gain associated with (3). A set  $\mathbb{X}_f$  with these properties can be calculated with the approach proposed in [13].

By eliminating the state variables the optimal control problem (2) can be stated in the compact form

$$\begin{aligned} \min_U \quad & V(x, U) \\ \text{s. t.} \quad & GU - Ex \leq w, \end{aligned} \quad \mathbb{P}(x)$$

with  $V(x, U) = \frac{1}{2}x'Yx + x'FU + \frac{1}{2}U'HU$  and  $Y \in \mathbb{R}^{n \times n}$ ,  $F \in \mathbb{R}^{n \times mN}$ ,  $H \in \mathbb{R}^{mN \times mN}$ ,  $H \succ 0$ ,  $G \in \mathbb{R}^{q \times mN}$ ,  $w \in \mathbb{R}^q$ ,  $E \in \mathbb{R}^{q \times n}$ , where  $q$  denotes the number of constraints [1]. Since  $H$  is positive definite,  $V(x, U)$  is a strictly convex function of  $U$  for any fixed  $x$ . The constraints define a convex polytope in  $U$  for any fixed  $x$ . Consequently,  $\mathbb{P}(x)$  has a unique solution if there exists any  $U$  that respects the constraints. Let  $\mathcal{X} \subseteq \mathbb{X}$  be the set of states such that  $\mathbb{P}(x)$  has a solution. We refer by  $U^* : \mathcal{X} \rightarrow \mathbb{U}^N$  to the optimal solution of  $\mathbb{P}(x)$  and by  $V^* : \mathcal{X} \rightarrow \mathbb{R}$  to the corresponding optimal value function. Formally,

$$\begin{aligned} U^*(x) = \arg \min_U \quad & V(x, U) \\ \text{s. t.} \quad & GU - Ex \leq w, \end{aligned} \quad (5)$$

and  $V^*(x) = V(x, U^*(x))$ . Let  $u^* : \mathcal{X} \rightarrow \mathbb{U}$  denote the first  $m$  elements of the optimal  $U^*(x)$ . Applying  $u^*(x)$  results in the predicted successor state  $x^+ = Ax + Bu^*(x)$ .

For later use we note that the solution of the optimization problem  $\min_U V(x, U)$ , i.e.  $\mathbb{P}(x)$  without constraints, is given by

$$U^*(x) = -H^{-1}F'x, \quad (6)$$

which immediately follows from the optimality conditions. Finally, we note that the constraints of  $\mathbb{P}(x)$  define a convex polytope in the combined space of states and inputs

$$\hat{\mathcal{P}} = \{(U, x) \in \mathbb{R}^{mN} \times \mathbb{R}^n \mid GU - Ex \leq w\}, \quad (7)$$

which is full dimensional, bounded and contains the origin in its interior if the assumptions on  $\mathbb{X}$  and  $\mathbb{U}$  hold.

### 2.1. Notation and preliminaries

For any matrix  $M \in \mathbb{R}^{q \times t}$ , let  $M^i$  and  $M^{\mathcal{W}}$  be the row vector and sub-matrix of row vectors indicated by  $i \in \{1, \dots, q\}$  and the ordered subset  $\mathcal{W} \subseteq \{1, \dots, q\}$ , respectively. Let  $\kappa(M) = \frac{\lambda_{\max}(M)}{\lambda_{\min}(M)}$  denote the condition number of any quadratic positive definite matrix  $M$ , where  $\lambda_{\max}(M)$  and  $\lambda_{\min}(M)$  refer to the largest and smallest eigenvalue of  $M$ , respectively.

For an arbitrary but fixed state  $x$  we call a point  $U \in \mathbb{R}^{mN}$  *feasible* for  $\mathbb{P}(x)$ , if the pair  $(x, U)$  fulfills the constraints in  $\mathbb{P}(x)$ . We call the quadratic program  $\mathbb{P}(x)$  *feasible* if there exists at least one feasible  $U$  for the given  $x$ . The  $i$ -th constraint in  $\mathbb{P}(x)$  is called *inactive* at the optimum  $U^*(x)$ , if  $G^i U^*(x) < w^i + E^i x$ , and *active* if  $G^i U^*(x) = w^i + E^i x$ . The index sets of active and inactive constraints for a state  $x \in \mathcal{X}$  are denoted by

$$\begin{aligned} \mathcal{A}(x) &= \{i \in \mathcal{Q} \mid G^i U^*(x) = w^i + E^i x\}, \\ \mathcal{I}(x) &= \{i \in \mathcal{Q} \mid G^i U^*(x) < w^i + E^i x\}, \end{aligned} \quad (8)$$

respectively, where  $\mathcal{Q} = \{1, \dots, q\}$ . Bemporad et al. [1] showed that  $U^* : \mathcal{X} \rightarrow \mathbb{U}^N$  introduced in (5) is a continuous piecewise affine function. More precisely, there exist  $n_{\mathcal{P}}$  gains  $\bar{K}_i \in \mathbb{R}^{mN \times n}$ , biases  $\bar{b}_i \in \mathbb{R}^{mN}$  and polytopes  $\mathcal{P}_i$  such that

$$U^*(x) = \begin{cases} \bar{K}_1 x + \bar{b}_1 & \text{if } x \in \mathcal{P}_1, \\ \vdots & \vdots \\ \bar{K}_{n_{\mathcal{P}}} x + \bar{b}_{n_{\mathcal{P}}} & \text{if } x \in \mathcal{P}_{n_{\mathcal{P}}} \end{cases} \quad (9)$$

is continuous and  $\cup_{i=1, \dots, n_{\mathcal{P}}} \mathcal{P}_i = \mathcal{X}$  and  $\text{int}(\mathcal{P}_i) \cap \text{int}(\mathcal{P}_j) = \emptyset$  for all  $i \neq j$ .

We refer to calculations that are carried out before the runtime of the model predictive controller as *offline* calculations, and to calculations that are necessary during runtime as *online* calculations.

Given the considered MPC problem (2), the following results hold true.

**Proposition 1.** *There exist strictly positive constants  $a_1$ ,  $a_2$ ,  $a_3$  such that  $x \in \mathcal{X}$  implies*

$$\begin{aligned} (i) \quad & a_1 \|x\|_2^2 \leq V^*(x) \leq a_2 \|x\|_2^2, \\ (ii) \quad & V^*(x^+) - V^*(x) \leq -a_3 \|x\|_2^2, \end{aligned}$$

*i.e. the optimal value function  $V^*(x)$  is a Lyapunov function for the closed-loop system (1).*

We omit a proof of this statement and refer the reader to the literature (see [14, §2.5.3.1, p. 142], [14, Prop. 2.17, p. 119] for a rigorous proof; see also [15]). Note that exponential stability of the closed-loop system follows from Proposition 1 (see [14, Thm. 2.24, p. 123] for a proof for the particular choices of  $\mathbb{X}_f$  and  $P$ ).

### 3. Constraint removal in linear model predictive control

It is the central idea of the proposed method to remove constraints that are inactive at the optimal solution from the quadratic program  $\mathbb{P}(x)$  *before* solving it. We show below that the number of active constraints is a nonincreasing function of time for the controlled system. Consequently, more and more constraints can be dropped as the controller drives the system towards the origin.

We state the reduced optimal control problem more precisely in Sec. 3.1 and show that it has the same solution as the original problem  $\mathbb{P}(x)$ . In Sections 3.2 and 3.3 we describe two methods for the detection of inactive constraints.

#### 3.1. Reduced optimal control problem

Let  $x \in \mathcal{X}$  be arbitrary and assume we know that some constraints in  $\mathbb{P}(x)$  are inactive at the optimum before having solved  $\mathbb{P}(x)$ . Proposition 2 states that these constraints are irrelevant for the optimal solution and can be removed from  $\mathbb{P}(x)$ .

**Proposition 2.** *Let  $x \in \mathcal{X}$  be arbitrary and let  $\tilde{\mathcal{I}} \subseteq \mathcal{I}(x)$  be an arbitrary subset of the inactive constraints. Consider the reduced optimization problem*

$$\begin{aligned} \min_{\hat{U}} \quad & V(x, \hat{U}) \\ \text{s. t.} \quad & G^{\mathcal{Q} \setminus \tilde{\mathcal{I}}} \hat{U} - E^{\mathcal{Q} \setminus \tilde{\mathcal{I}}} x \leq w^{\mathcal{Q} \setminus \tilde{\mathcal{I}}}. \end{aligned} \quad \hat{\mathbb{P}}(x)$$

Then  $\hat{\mathbb{P}}(x)$  has a unique solution, which we denote by  $\hat{U}^*(x)$ . This solution is equal to the solution obtained from  $\mathbb{P}(x)$ , i.e.  $\hat{U}^*(x) = U^*(x)$  and  $V(x, \hat{U}^*(x)) = V(x, U^*(x)) = V^*(x)$ .

*Proof.* Assume  $\hat{U}^*(x) \neq U^*(x)$  and show a contradiction results. Since the feasible region of problem  $\hat{\mathbb{P}}(x)$  is a superset of that of  $\mathbb{P}(x)$ , it follows that  $U^*(x)$  is a feasible point for  $\hat{\mathbb{P}}(x)$ . Since  $\hat{\mathbb{P}}(x)$  is a strictly convex QP, there holds:

$$V(x, \hat{U}^*(x)) < V(x, U^*(x)) \quad (10)$$

Given that  $\tilde{\mathcal{I}} \subseteq \mathcal{I}(x)$ , it follows that

$$G^{\tilde{\mathcal{I}}} U^*(x) - E^{\tilde{\mathcal{I}}} x < w^{\tilde{\mathcal{I}}}. \quad (11)$$

Since both  $\hat{U}^*(x)$  and  $U^*(x)$  are feasible points of  $\hat{\mathbb{P}}$ , we have

$$G^{\mathcal{Q} \setminus \tilde{\mathcal{I}}} U^*(x) - E^{\mathcal{Q} \setminus \tilde{\mathcal{I}}} x \leq w^{\mathcal{Q} \setminus \tilde{\mathcal{I}}}, \quad G^{\mathcal{Q} \setminus \tilde{\mathcal{I}}} \hat{U}^*(x) - E^{\mathcal{Q} \setminus \tilde{\mathcal{I}}} x \leq w^{\mathcal{Q} \setminus \tilde{\mathcal{I}}}. \quad (12)$$

Now define  $d = \hat{U}^*(x) - U^*(x)$ . By convexity of  $V(x, \cdot)$  and (10) it follows that

$$\begin{aligned} V(x, U^*(x) + \alpha d) &\leq (1 - \alpha)V(x, U^*(x)) + \alpha V(x, \hat{U}^*(x)) \\ &= V(x, U^*(x)) - \alpha \left( V(x, U^*(x)) - V(x, \hat{U}^*(x)) \right) < V(x, U^*(x)). \end{aligned} \quad (13)$$

for any  $\alpha \in (0, 1]$ . From (11) and (12), there exists some  $\alpha^* \in (0, 1]$  such that

$$G^{\mathcal{Q}}(U^*(x) + \alpha^* d) - E^{\mathcal{Q}}x \leq w^{\mathcal{Q}}. \quad (14)$$

From (13) and (14), it follows that  $U^*(x)$  is not the solution to problem  $\mathbb{P}(x)$ , which completes the proof.  $\blacksquare$

### 3.2. Constraint removal based on regions of activity

In this section we show how to use the so-called regions of activity [10] to detect inactive constraints. The region of activity  $\mathcal{G}_i$  is defined as the subset of  $\mathcal{X}$  in which the constraint  $i$  is active, i.e.  $\mathcal{G}_i = \{x \in \mathcal{X} \mid i \in \mathcal{A}(x)\}$ , or equivalently

$$\mathcal{G}_i = \{x \in \mathcal{X} \mid G^i U^*(x) - E^i x = w^i\}. \quad (15)$$

By definition of the region of activity we have

$$i \in \mathcal{A}(x) \text{ if and only if } x \in \mathcal{G}_i. \quad (16)$$

An example is shown in Fig. 1a.

Assuming the regions of activity have been calculated *offline*, we can use them *online* to detect which constraints are active respectively inactive by testing whether  $x \in \mathcal{G}_i$  for all  $i \in \mathcal{Q}$  for the current  $x$ . If we identify and remove *all* inactive constraints, a reduced quadratic program  $\hat{\mathbb{P}}(x)$  with only equality constraints remains, which is considerably simpler to solve than the general case  $\hat{\mathbb{P}}(x)$  (see e.g. [17, p. 164, Sec. 5.1.4.2]).

The following lemma summarizes the relation of the regions of activity  $\mathcal{G}_i$  to the polytopes  $\mathcal{P}_j$  of the explicit solution (9).

**Lemma 3.** *Let  $i \in \mathcal{Q}$ . The region of activity  $\mathcal{G}_i$  defined in (15) is the union of a finite number of the polytopes  $\mathcal{P}_k$  from (9), i.e. there exists a set  $\mathcal{K}_i \subseteq \{1, \dots, n_{\mathcal{P}}\}$  such that*

$$\mathcal{G}_i = \bigcup_{k \in \mathcal{K}_i} \mathcal{P}_k. \quad (17)$$

*Proof.* We prove the claim by constructing  $\mathcal{K}_i$ . The sets of active and inactive constraints are constant on the interior of a polytope  $\mathcal{P}_i$ , i.e. we have  $\mathcal{A}(x) = \mathcal{A}(\bar{x})$  and  $\mathcal{I}(x) = \mathcal{I}(\bar{x})$  for any  $x, \bar{x} \in \mathcal{P}_i$  [1, 18]. It is therefore meaningful to denote the set of constraints that are active (resp. inactive) on the interior of  $\mathcal{P}_i$  by  $\mathcal{A}_i$  (resp.  $\mathcal{I}_i$ ). The index set  $\mathcal{K}_i$  is then given by

$$\mathcal{K}_i = \{j \in \{1, \dots, n_{\mathcal{P}}\} \mid i \in \mathcal{A}_j\},$$

which proves the claim.  $\blacksquare$

The union of a finite number of polytopes is in general not convex and not connected. Fig. 1a shows an example of a non-convex region of activity. Due to nonconvexity, it is in general computationally expensive to check whether

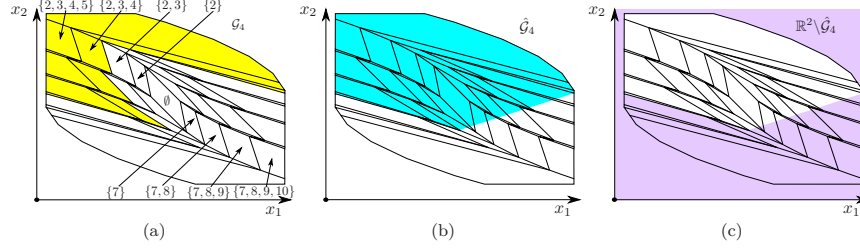


Figure 1: (a) Region of activity for a double integrator example.  $\mathcal{G}_4$  (yellow) is the union of all polytopes in which constraint 4 is active. Note  $\mathcal{G}_4$  is not convex. (b) The convex hull  $\hat{\mathcal{G}}_4$  (cyan) is a convex outer approximation of the region of activity  $\mathcal{G}_4$ . (c) Constraint 4 is inactive for all states  $x \notin \hat{\mathcal{G}}_4$  (cf. Lemma 4).

$x \in \mathcal{G}_i$ . We overcome this problem by using convex outer approximations  $\hat{\mathcal{G}}_i$  of the regions of activity  $\mathcal{G}_i$ . This was first suggested in [10, 11]. The equivalence (16) no longer holds for convex outer approximations  $\hat{\mathcal{G}}_i \supset \mathcal{G}_i$  but has to be replaced by

$$i \in \mathcal{A}(x) \text{ implies } x \in \hat{\mathcal{G}}_i.$$

This implication can be used as follows.

**Proposition 4.** *Let  $x \in \mathcal{X}$  be arbitrary, let  $i \in \mathcal{Q}$  be an arbitrary constraint and let  $\mathcal{G}_i$  be as in (15). Consider any outer approximation  $\hat{\mathcal{G}}_i \supset \mathcal{G}_i$  of the region of activity  $\mathcal{G}_i$ . Then*

$$x \notin \hat{\mathcal{G}}_i \text{ implies } i \in \mathcal{I}(x),$$

*i.e. constraint  $i$  is inactive at the optimal solution to  $\mathbb{P}(x)$ .*

*Proof.* Proposition 4 can be shown by contradiction. Assume there exists an  $x \notin \hat{\mathcal{G}}_i$  such that  $i \in \mathcal{A}(x)$ . According to (16)  $i \in \mathcal{A}(x)$  implies  $x \in \mathcal{G}_i$ . From  $\hat{\mathcal{G}}_i \supset \mathcal{G}_i$  we infer  $x \in \hat{\mathcal{G}}_i$ , which is a contradiction. ■

Proposition 4, which is illustrated in Fig. 1, can be used to identify inactive constraints online by testing whether  $x \in \hat{\mathcal{G}}_i$  for all  $i \in \mathcal{Q}$ . These set membership tests can be carried out efficiently for *convex* outer approximations  $\hat{\mathcal{G}}_i$  [11]. We use ellipsoidal and hyperrectangular convex outer approximations of the regions of activity in Sec. 4.

### 3.3. Constraint removal based on precalculated bounds on the optimal value function

This section presents a method for the detection of inactive constraints that does not require to determine the regions of activity  $\mathcal{G}_i$  or any approximations  $\hat{\mathcal{G}}_i$ . Essentially, we show a constraint  $i \in \mathcal{Q}$  is inactive, whenever the optimal value function  $V^*(x)$  of  $\mathbb{P}(x)$  drops below a certain bound  $\sigma_i^*$ , where  $\sigma_i^*$  does not depend on  $x$  and therefore may be calculated offline. If the optimal value



function  $V^*(x)$  of  $\mathbb{P}(x)$  is a Lyapunov function for the controlled system,  $V^*(x)$  strictly decreases along any trajectory, and therefore constraint  $i$  remains inactive once the optimal value function dropped below  $\sigma_i^*$  while solving  $\mathbb{P}(x)$  online.

We first introduce the bounds  $\sigma_i^*$  informally and state their properties more precisely below. Consider an arbitrary  $i \in \mathcal{Q}$  and choose  $x \in \mathcal{X}$  such that  $i \in \mathcal{A}(x)$ , i.e. let  $x$  be such that constraint  $i$  is active at the optimal solution of  $\mathbb{P}(x)$ . Let  $\mathcal{Q} \setminus i$  be short for the set difference  $\mathcal{Q} \setminus \{i\}$ . If constraint  $i$  is active, we can turn it into an equality constraint in  $\mathbb{P}(x)$ , consequently the QP

$$\begin{aligned} \min_U \quad & V(x, U) \\ \text{s. t.} \quad & G^{\mathcal{Q} \setminus i} U - E^{\mathcal{Q} \setminus i} x \leq w^{\mathcal{Q} \setminus i}, \\ & G^i U - E^i x = w^i, \end{aligned} \tag{18}$$

has the same unique solution as  $\mathbb{P}(x)$  in this case. If we pose the same optimization problem as (18) but minimize with respect to both  $x$  and  $U$ , we obtain a number that is a lower bound for  $V(x, U^*(x))$  on  $\mathcal{G}_i$ . We call this bound  $\sigma_i^*$ :

$$\begin{aligned} \sigma_i^* := \min_{x, U} \quad & V(x, U) \\ \text{s. t.} \quad & G^{\mathcal{Q} \setminus i} U - E^{\mathcal{Q} \setminus i} x \leq w^{\mathcal{Q} \setminus i}, \\ & G^i U - E^i x = w^i, \\ & x \in \mathcal{X}. \end{aligned} \tag{19}$$

Note that (19) may be infeasible. This case occurs if constraint  $i$  is not active in  $\mathbb{P}(x)$  for any  $x \in \mathcal{X}$ , for example, because other constraints than  $i$  are more restrictive (see e.g. [19, Sec. 4.1.1, p. 128] or [20, Def. 5, p. 492] on redundant constraints). It proves useful to formally extend the definition of  $\sigma_i^*$  in (19) by

$$\sigma_i^* := \infty \text{ if (19) is infeasible.} \tag{20}$$

Before showing how to use the bounds  $\sigma_i^*$  to detect inactive constraints, we summarize some properties of (19) in Lemma 5.

**Lemma 5.** *Let  $i \in \mathcal{Q}$  be arbitrary and consider the QP (19). The following statements hold:*

- (i) *If QP (19) is feasible, it has a unique solution.*
- (ii) *If (19) is feasible, then  $0 < \sigma_i^* < \infty$ .*
- (iii) *If (19) is infeasible, then constraint  $i$  is always inactive in  $\mathbb{P}(x)$ , or equivalently  $i \in \mathcal{I}(x)$  for all  $x \in \mathcal{X}$ .*

*Proof.* (i) The claim follows from Proposition 1 on  $\mathbb{P}(x)$ , which can be seen as follows. The cost function of  $\mathbb{P}(x)$  can be rewritten as

$$V(x, U) = \frac{1}{2} \begin{pmatrix} x' & U' \end{pmatrix} \begin{pmatrix} Y & F \\ F' & H \end{pmatrix} \begin{pmatrix} x \\ U \end{pmatrix}. \tag{21}$$

According to Proposition 1, there exists an  $a_1 > 0$  such that  $V^*(x) \geq a_1 \|x\|_2^2$  for all  $x \in \mathcal{X}$ . Together with  $V^*(x) = V(x, U^*(x))$  this implies  $V(x, U^*(x)) > 0$  for all  $x \in \mathcal{X} \setminus \{0\}$ . Because  $V(x, U^*(x)) \leq V(x, U)$  by definition of  $U^*(x)$ , we also have  $V(x, U) > 0$  for all  $x \in \mathcal{X} \setminus \{0\}$  and all feasible  $U$ , or equivalently for all  $(U, x) \in \hat{\mathcal{P}}$  with  $x \neq 0$ , where  $\hat{\mathcal{P}}$  is as defined in (7). Therefore,  $V(x, U)$  is positive definite on a full dimensional polytope that contains the origin (i.e. on  $\hat{\mathcal{P}}$ ). This implies the matrix in (21) is positive definite. Since the constraints define a convex polytope, the claim holds.

(ii) We already established that the cost function in (19) is positive definite. Consequently,  $\sigma_i^* \geq 0$ , and it remains to show  $\sigma_i^* \neq 0$ . We assume there exists an  $i \in \{1, \dots, q\}$  such that  $\sigma_i^* = 0$  in (19) and show that a contradiction results. From (21) we infer that  $\sigma_i^* = 0$  implies  $(U, x) = (0, 0)$ . By assumption, however,  $(U, x) = (0, 0)$  lies in the interior of  $\hat{\mathcal{P}}$ . According to the definition of  $\hat{\mathcal{P}}$  in (7)  $x \in \text{int}(\hat{\mathcal{P}})$  implies none of the constraints  $G^j U - E^j x \leq w^j$ ,  $j = 1, \dots, q$  are active at the point  $(U, x) = (0, 0)$ . Equivalently,  $G^j U - E^j x - w^j < 0$  holds for all  $j = 1, \dots, q$ . This contradicts the equality constraint  $G^i U - E^i x = w^i$  in (19), however.

(iii) Since (19) is infeasible, there exists no  $(U, x) \in \mathbb{R}^{mN} \times \mathcal{X}$  such that  $G^{\mathcal{Q} \setminus i} U - E^{\mathcal{Q} \setminus i} x \leq w^{\mathcal{Q} \setminus i}$  and  $G^i U - E^i x = w^i$ . In particular  $(U^*(x), x)$  does not fulfill  $G^{\mathcal{Q} \setminus i} U^* - E^{\mathcal{Q} \setminus i} x \leq w^{\mathcal{Q} \setminus i}$  and  $G^i U^*(x) - E^i x = w^i$  for any  $x \in \mathcal{X}$ . This implies  $G^i U^*(x) - E^i x < w^i$  for all  $x \in \mathcal{X}$ , which in turn implies  $i \in \mathcal{I}(x)$  for all  $x \in \mathcal{X}$  according to the definition of  $\mathcal{I}(x)$  in (8). ■

The bounds  $\sigma_i^*$  can be used as follows to detect inactive constraints.

**Proposition 6.** *Let  $i \in \mathcal{Q}$  and  $x \in \mathcal{X}$  be arbitrary, and let  $\sigma_i^*$  be defined as in (19) and (20). If  $V^*(x) < \sigma_i^*$ , then constraint  $i$  is inactive at the optimal solution to  $\mathbb{P}(x)$ , i.e.  $i \in \mathcal{I}(x)$ .*

*Proof.* We consider the cases  $\sigma_i^* = \infty$  and  $\sigma_i^* < \infty$  separately. If  $\sigma_i^* = \infty$  we have  $i \in \mathcal{I}(x)$  for any  $x \in \mathcal{X}$  according to Lemma 5. The case  $\sigma_i^* < \infty$  can be proven by contradiction. Assume the claim does not hold, i.e. there exists an  $x \in \mathcal{X}$  such that  $V^*(x) < \sigma_i^*$  and  $i \notin \mathcal{I}(x)$ , or equivalently

$$V^*(x) < \sigma_i^* \text{ and } i \in \mathcal{A}(x), \quad (22)$$

where we used  $\mathcal{A}(x) \cup \mathcal{I}(x) = \mathcal{Q}$  and  $\mathcal{I}(x) \cap \mathcal{A}(x) = \emptyset$ , which follows from (8). From (22) we infer that  $(U^*(x), x)$  satisfies  $G^{\mathcal{Q} \setminus i} U^*(x) - E^{\mathcal{Q} \setminus i} x \leq w^{\mathcal{Q} \setminus i}$  and  $G^i U^*(x) - E^i x = w^i$ , which implies  $(U^*(x), x)$  is a feasible point for (19). Since  $(x, U^*(x))$  is feasible but not necessarily an optimal point for (19), we have  $\sigma_i^* \leq V(x, U^*(x))$ , which contradicts the assumption  $V^*(x) < \sigma_i^*$ . ■

The proof of Prop. 6 assumes  $\mathbb{P}(x)$  has been solved and  $U^*(x)$  is known. Once  $\mathbb{P}(x)$  has been solved for a particular  $x \in \mathcal{X}$ , the sets  $\mathcal{A}(x)$  and  $\mathcal{I}(x)$  are known, therefore there is no need to apply Prop. 6 to detect inactive constraints. If we combine Prop. 6 with the Lyapunov property of  $V^*(x)$ , however, we can infer a constraint  $i$  to remain inactive for all  $t$  once  $V^*(x)$  dropped below  $\sigma_i^*$ .

Consequently, it can be removed from  $\mathbb{P}(x)$ . This is stated more precisely in the following corollary.

**Corollary 7.** *Consider the optimization problem  $\mathbb{P}(x(t_0))$  at time step  $t_0$  with state  $x(t_0) \in \mathcal{X}$ . Let  $i \in \mathcal{Q}$  be arbitrary, and let  $\sigma_i^*$  be defined as in (19) and (20). Then*

$$V^*(x(t_0)) < \sigma_i^* \text{ implies } i \in \mathcal{I}(x(t)) \text{ for all } t \geq t_0,$$

*i.e. constraint  $i$  is inactive along the trajectory of the controlled system for all  $t \geq t_0$ .*

*Proof.* According to Proposition 1 (ii),  $V^*(x)$  is nonincreasing along any trajectory of the closed-loop system. Together with  $V^*(x(t_0)) < \sigma_i^*$  this yields  $V^*(x(t)) < \sigma_i^*$  for all  $t \geq t_0$  for the closed-loop system. Proposition 6 implies  $i \in \mathcal{I}(x(t))$  for all  $t \geq t_0$  for the closed-loop system. ■

Both Prop. 4 and Cor. 7 provide criteria for detecting inactive constraints in  $\mathbb{P}(x)$ . The criterion from Cor. 7 is computationally simpler, since it only requires comparing two real numbers for each constraint. In contrast, a set membership test has to be carried out in Prop. 4 for each constraint. Moreover, the criterion from Cor. 7 needs only be carried out for each  $i \in \mathcal{Q}$ , until constraint  $i \in \mathcal{Q}$  is detected to be inactive for the first time. In contrast, a constraint that is detected to be inactive with Prop. 4 at time  $t_k$  does in general not remain inactive for all  $t \geq t_k$ . This difference is obvious from the examples presented in Sec. 4.

Proposition 1 implies that  $V^*(x) \rightarrow 0$  as  $t \rightarrow \infty$ . Since we have  $\sigma_i^* > 0$  for all  $i$ , every constraint will turn inactive at some point along any trajectory of the closed-loop system. As a consequence, the number of constraints goes to zero until only the unconstrained optimization problem remains. This is summarized in the following remark for ease of reference.

**Remark 8.** *Consider the reduced optimization problem  $\tilde{\mathbb{P}}(x(t))$  at time step  $t$  with state  $x(t)$ , and let  $\tilde{\mathcal{I}}(x(t))$  result from applying Prop. 6 to all  $i \in \mathcal{Q}$ . Then the number of constraints  $|\mathcal{Q} \setminus \tilde{\mathcal{I}}(x(t))|$  in  $\tilde{\mathbb{P}}(x)$  is a nonincreasing function of  $t$  and goes to zero along any trajectory of the closed-loop system as  $t \rightarrow \infty$ .*

We conclude Sec. 3.3 with an illustration of the bounds  $\sigma_i^*$  in Figure 2. A geometric interpretation of the constraints in (18) is required for this purpose. The equality constraint in (18) defines the linear subspace

$$\mathcal{H}^i = \left\{ \begin{pmatrix} U \\ x \end{pmatrix} \in \mathbb{R}^{mN+n} \left| (G^i - E^i) \begin{pmatrix} U \\ x \end{pmatrix} = w^i \right. \right\}.$$

Similarly, the constraints  $G^{\mathcal{Q} \setminus i} U - E^{\mathcal{Q} \setminus i} x \leq w^{\mathcal{Q} \setminus i}$  define the polytope

$$\mathcal{P}^{\mathcal{Q} \setminus i} = \left\{ \begin{pmatrix} U \\ x \end{pmatrix} \in \mathbb{R}^{mN+n} \left| (G^{\mathcal{Q} \setminus i} - E^{\mathcal{Q} \setminus i}) \begin{pmatrix} U \\ x \end{pmatrix} \leq w^{\mathcal{Q} \setminus i} \right. \right\}.$$

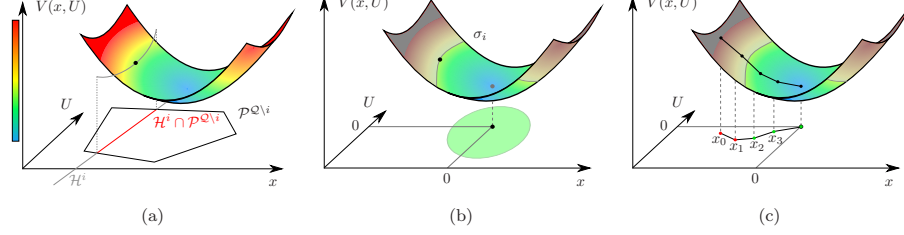


Figure 2: Geometric interpretation of  $\sigma_i^*$  and the detection of inactive constraints described in Cor. 7. The figure sketches a hypothetical system with  $n = 1$  and  $mN = 1$ . See the end of Sec. 3.2 for an explanation.

---

**Algorithm 1** MPC algorithm with constraint removal. Summary of calculations to find the control law for the successor state  $x^+ = Ax + Bu^*(x)$ .

---

- 1: **Input:**  $x^+, V^*(x), \mathcal{R} = \mathcal{Q} \setminus \tilde{\mathcal{I}}(x), \sigma_i^*, i \in \mathcal{R}$ .
  - 2: Set  $\tilde{\mathcal{I}}(t) = \emptyset$ .
  - 3: **for all**  $i \in \mathcal{R}$  **do**
  - 4:   **if**  $V^*(x) < \sigma_i^*$  **then**
  - 5:     Constraint  $i$  is inactive at state  $x^+$ :  $\tilde{\mathcal{I}} \leftarrow \tilde{\mathcal{I}} \cup \{i\}$ .
  - 6:   **end if**
  - 7: **end for**
  - 8: Remove inactive constraints:  $\mathcal{R} \leftarrow \mathcal{R} \setminus \tilde{\mathcal{I}}$ .
  - 9: **if**  $\mathcal{R} = \emptyset$  **then**
  - 10:   QP is unconstrained and the solution is  $U^*(x^+) = -H^{-1}F'x^+$ .
  - 11: **else**
  - 12:   Solve reduced QP  $\tilde{\mathbb{P}}(x^+)$  with constraint set  $\mathcal{R}$  for  $U^*(x^+)$ .
  - 13: **end if**
  - 14: **Output:**  $U^*(x^+), V^*(x^+), \mathcal{R}$ .
- 

Figure 2a illustrates  $\mathcal{H}^i$ ,  $\mathcal{P}^{Q \setminus i}$  and their intersection  $\mathcal{H}^i \cap \mathcal{P}^{Q \setminus i}$  (highlighted in red) for a hypothetical example. The bound  $\sigma_i^*$  results from optimizing  $V(x, U)$  on  $\mathcal{H}^i \cap \mathcal{P}^{Q \setminus i}$ . Corollary 7 essentially states that constraint  $i$  is inactive on the sublevel-set  $\{(U, x) | V(x, U) < \sigma_i^*\}$ , which is illustrated in Fig. 2b. Since  $V^*(x)$  is a Lyapunov function and therefore nonincreasing, this sublevel-set is positive invariant for the controlled system (cf. Fig. 2c).

#### 4. Examples

We apply the approach presented in this paper to two examples summarized in Table 1. More detailed descriptions of the systems are given in Appendix A. For both examples we used the maximal constraint admissible set as terminal set  $\mathbb{X}_f$ , which we construct with the approach presented in [13]. We compare the computation times required to solve the full quadratic program  $\mathbb{P}(x)$  and

the reduced quadratic program  $\hat{\mathbb{P}}(x)$ , where several different methods are used to detect the inactive constraints removed from  $\hat{\mathbb{P}}(x)$ . Specifically, we compare the following MPC implementations:

- **full-QP:**  $\mathbb{P}(x)$  without any reduction.
- **ell-QP:** Inactive constraints are detected with Prop. 4, and the reduced quadratic program  $\hat{\mathbb{P}}(x)$  is solved. The outer approximations  $\hat{\mathcal{G}}_i$  of the regions of activity are ellipsoids [11].
- **hyp-QP:** Same as ell-QP, but the outer approximations  $\hat{\mathcal{G}}_i$  of the regions of activity are hyperrectangles [11].
- **lyap-QP:** Inactive constraints are detected with Cor. 7, and the reduced QP  $\hat{\mathbb{P}}(x)$  is solved.

We randomly chose feasible initial conditions for both examples and calculated trajectories of the closed-loop system until  $|x(t)| \leq 10^{-3}$  in all cases. The number of initial conditions and the number of solved QPs are given in Tab. 1.

The results of the simulations are shown in Fig. 3 in terms of the cumulative distribution function (cdf)  $h_{\text{cdf}}(t_{\text{MPC}})$ , where  $t_{\text{MPC}}$  is the time required to find the control action by solving  $\mathbb{P}(x)$  or  $\hat{\mathbb{P}}(x)$ , respectively. The cdf  $h_{\text{cdf}}(t)$  equals the fraction of cases in which  $U^*(x)$  was found in computational time  $t$  or less. We stress all reported times include the computational times required to detect inactive constraints where applicable. More precisely, times reported for ell-QP and hyp-QP include the computational times required to test whether  $x \in \hat{\mathcal{G}}_i$  for all  $i \in \mathcal{Q}$ , to set up the reduced QP  $\hat{\mathbb{P}}(x)$ , and to solve it. Times reported for lyap-QP include the computational times required to carry out the tests  $V^*(x) < \sigma_i^*$  for all  $i \in \mathcal{R}$ , to set up the reduced QP  $\hat{\mathbb{P}}(x)$ , and to solve it.

Consider the results for the double integrator example in Fig. 3a first. Lyap-QP is faster than all other approaches by an order of magnitude for about 85% of all cases (leftmost shoulder of the purple curve in Fig. 3a). Lyap-QP is faster in these cases, because it detects the QP to be unconstrained. This implies no QP solver is required at all, because the optimal solution is given by (6), cf. line 10 in Algorithm 1. Lyap-QP is always faster than full-QP, ell-QP is faster than full-QP in about 95% of all cases, but slower in the remaining 5%. Hyp-QP and full-QP result in very similar cumulative distribution functions. Note that lyap-QP is faster than hyp-QP and ell-QP for times smaller than about  $4.5 \cdot 10^{-3}$ s. For  $t_{\text{MPC}}$  larger than  $4.5 \cdot 10^{-3}$ s lyap-QP is still faster, but the difference is negligible.

Example	$n$	$m$	$N$	$q$	$\# x_0$	$\# \text{ QPs solved}$
Double integrator	2	1	2	62	7,500	1,376,233
MIMO system	10	3	30	898	6,800	1,127,679

Table 1: Summary of the examples.

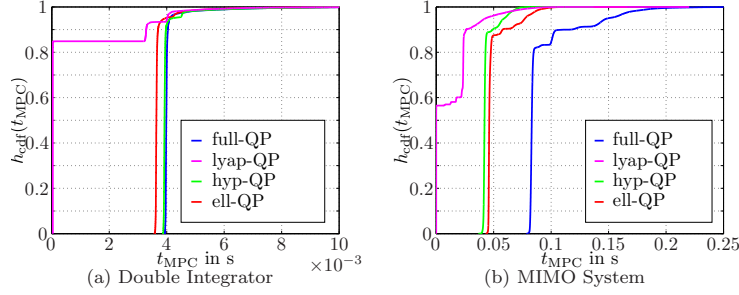


Figure 3: Cumulative distribution functions (cdf) of the computation times. The leftmost shoulder of the purple curve is at  $5.8 \cdot 10^{-5}$ s and  $2.0 \cdot 10^{-4}$ s in (a) and (b), respectively.

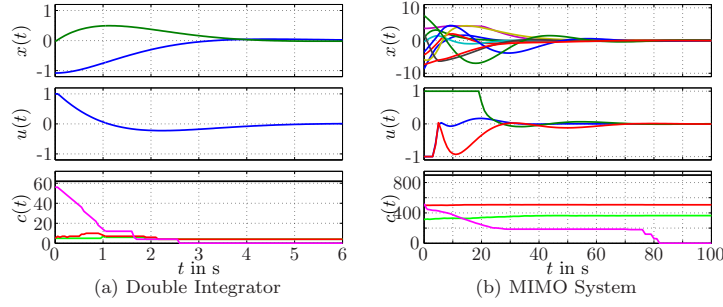


Figure 4: States  $x_i(t)$ ,  $i = 1, \dots, n$ , inputs  $u_i(t)$ ,  $i = 1, \dots, m$  and the number of constraints  $c(t)$  along the closed-loop trajectories for a random initial state. Black, red, green and magenta curves in the subfigures for  $c(t)$  correspond to full-QP, ell-QP, hyp-QP and lyap-QP, respectively.

Consider the results of the MIMO system presented in Fig. 3b next. About 55% of the QPs are detected to be unconstrained by lyap-QP. The solution can be obtained from (6) and therefore is found in less than  $2.0 \cdot 10^{-4}$ s (leftmost shoulder of the purple curve in Fig. 3b). More importantly, in about 95% of the QPs the control law can be found faster with lyap-QP than in the *best* case with ell-QP, hyp-QP and full-QP. Lyap-QP leads by far to the best performance in this sense.

Figure 4 shows the state variables  $x_i(t)$ ,  $i = 1, \dots, n$ , the input variables  $u_i(t)$ ,  $i = 1, \dots, m$  and the number of constraints  $c(t)$  along the trajectory for a random initial point for both examples. Clearly, lyap-QP, hyp-QP and ell-QP are able to reduce the number of constraints in both cases.

According to Remark 8 the number of constraints for lyap-QP is nonincreasing along a trajectory of the closed-loop system. This is corroborated in Fig. 4 (magenta curves in a and b). Recall that a constraint which is detected to be inactive in lyap-QP will remain inactive, and thus the number of constraints goes to zero along the trajectory of the closed-loop system. This is not the case for ell-QP and hyp-QP: Even if the system state has reached the origin

the quadratic program to be solved is not detected to be unconstrained (cf. red curve for ell-QP, green curve for hyp-QP in Fig. 4a and b).

## 5. Conclusion

We proposed and analyzed a simple method for accelerating MPC for linear systems, linear constraints and quadratic objective functions. Essentially, we remove inactive constraints in the online MPC problem in every time step before solving it. Constraints can be detected to be inactive based on bounds on the MPC objective function. Because these bounds are independent of the current state, they can be calculated offline, i.e. once before actually starting the MPC controller (see Sec. 3.3). For an MPC problem with  $q$  constraints,  $q$  convex quadratic programs need to be solved offline (see (18)). Since each of these QPs belongs to the same complexity class as the online MPC problem  $\mathbb{P}(x)$  itself, the preparatory calculations are not a limitation. The additional operations required at runtime of the MPC controller are computationally inexpensive and they are clearly outweighed by the savings (see Sec. 4). The proposed method outperforms earlier ones that also used constraint removal, but identified inactive constraints with different techniques.

Future research will address the detection of inactive constraints under the presence of disturbances or plant-model mismatch, and the extension to nonlinear MPC.

## Acknowledgment

Support by the Deutsche Forschungsgemeinschaft (DFG) under grant MO 1086/11-1 is gratefully acknowledged.

## Appendix A. Details on the examples

*Double Integrator [20].* We consider the double integrator system that results from discretizing  $\ddot{y} = u$  with zero-order hold (ZOH) and a sample time of  $T_s = 0.05s$ . State and input constraints are  $-5 \leq x_1(t) \leq 5$ ,  $-0.5 \leq x_2(t) \leq 0.5$  and  $-1 \leq u(t) \leq 1$ ; and  $R = 1$ ,  $Q = \text{diag}(1, 10^{-3})$ .

This example is taken from [20]. The optimization problem is well-conditioned with  $\kappa(H) = 1.15$ .

*Multi-Input Multi-Output system [12].* We consider the state space system resulting from discretizing the continuous-time transfer function

$$G(s) = \begin{pmatrix} \frac{-5s+1}{36s^2+6s+1} & \frac{0.5}{8s+1} & 0 \\ 0 & \frac{0.1(-10s+1)}{s(8s+1)} & \frac{-0.1}{(64s^2+6s+1)s} \\ \frac{-2s+1}{12s^2+3s+1} & 0 & \frac{2(-5s+1)}{16s^2+2s+1} \end{pmatrix}, \quad (\text{A.1})$$

with ZOH and  $T_s = 1s$ . State space matrices are given in [21]. The state and input constraints are  $-10 \leq x_i(t) \leq 10$  for  $i \in (1, \dots, 10)$ ,  $-1 \leq u_j(t) \leq 1$

for  $j \in (1, \dots, 3)$ , respectively. The weighting matrices are  $Q = I^{n \times n}$  and  $R = 0.25I^{m \times m}$ . We reparameterize the input variables using a stabilizing LQR controller as proposed in [22] to obtain a well-conditioned optimization problem with  $\kappa(H) = 2.51$ .

## References

- [1] A. Bemporad, M. Morari, V. Dua, E. Pistikopoulos, The explicit linear quadratic regulator for constrained systems, *Automatica* 38 (2002) 3 – 20.
- [2] C. V. Rao, S. J. Wright, J. B. Rawlings, Application of interior-point methods to model predictive control, *J. Optim. Theor. Applic.* 99 (1998) 723–757.
- [3] Y. Wang, S. Boyd, Fast model predictive control using online optimization, *IEEE Trans. Contr. Syst. Techn.* 18 (2) (2010) 267–278.
- [4] S. Richter, C. N. Jones, M. Morari, Computational complexity certification for real-time mpc with input constraints based on the fast gradient method, *IEEE Trans. Auto. Contr.* 57 (6) (2012) 1391–1403.
- [5] B. O’Donoghue, G. Stathopoulos, , S. Boyd, A splitting method for optimal control, *IEEE Trans. Contr. Syst. Techn.* 21 (6) (2013) 2432–2442.
- [6] H. Ferreau, H. Bock, M. Diehl, An online active set strategy to overcome the limitations of explicit MPC, *International Journal of Robust and Nonlinear Control* 18 (2008) 816–830.
- [7] M. Zeilinger, C. Jones, M. Morari, Real-Time Suboptimal Model Predictive Control Using a Combination of Explicit MPC and Online Optimization, *IEEE Transactions on Automatic Control* 56 (7) (2011) 1524–1534.
- [8] G. Pannocchia, S. Wright, J. Rawlings, Partial enumeration MPC: Robust stability results and application to an unstable CSTR, *Journal of Process Control* 21 (10) (2011) 1459–1466.
- [9] G. Pannocchia, J. B. Rawlings, S. J. Wright, Fast, large-scale model predictive control by partial enumeration, *Automatica* 43 (2007) 852–860.
- [10] M. Jost, M. Mönnigmann, Accelerating online MPC with partial explicit information and linear storage complexity in the number of constraints, in: *Proceedings of the European Control Conference 2013, Zurich, Switzerland, 2013*, pp. 35–40.
- [11] M. Jost, M. Mönnigmann, Accelerating model predictive control by online constraint removal, in: *Proceedings of the Conference on Decision and Control 2013, Florence, Italy, 2013*, pp. 5764–5769.



- [12] M. Jost, G. Pannocchia, M. Mönnigmann, Online constraint removal: accelerating MPC with a Lyapunov function, *Automatica* 57 (2015) 164–169. doi:10.1016/j.automatica.2015.04.014.
- [13] E. G. Gilbert, K. Tin Tan, Linear systems with state and control constraints: The theory and application of maximal output admissible sets, *IEEE Transactions on Automatic Control* 36 (9) (1991) 1008–1020.
- [14] J. B. Rawlings, D. Q. Mayne, *Model Predictive Control: Theory and Design*, 1st Edition, Nob Hill Publishing, Madison, Wisconsin, USA, 2009.
- [15] D. Mayne, James B. Rawlings, Christopher V. Rao, Pierre O. M. Scokaert, Constrained model predictive control: Stability and optimality, *Automatica* 36 (2000) 789 – 814.
- [16] J. Nocedal, S. J. Wright, *Numerical Optimization*, 2nd Edition, Springer Verlag, 2006.
- [17] P. E. Gill, W. Murray, M. H. Wright, *Practical Optimization*, Elsevier, 2007.
- [18] A. Gupta, S. Bhartiya, P. Nataraj, A novel approach to multiparametric quadratic programming, *Automatica* 47 (9) (2011) 2112 – 2117.
- [19] S. Boyd, L. Vandenberghe, *Convex Optimization*, Cambridge University Press, 2009.
- [20] P. Tøndel, T. A. Johansen, A. Bemporad, An algorithm for multiparametric quadratic programming and explicit MPC solutions, *Automatica* 39 (2003) 489 – 497.
- [21] M. Jost, M. Mönnigmann, Minimal state space representation of an example system for online model predictive control, Tech. rep., Ruhr-Universität Bochum (2014).
- [22] J. A. Rossiter, B. Kouvaritakis, M. J. Rice, A numerically robust state-space approach to stable-predictive control strategies, *Automatica* 34 (1998) 65–73.